

멀티쓰레딩을 활용한 병행 점진 평가

한정란
 협성대학교 경영정보학과
 e-mail:jlhan@uhs.ac.kr

A Concurrent Incremental Evaluation Using Multithreading

Junglan Han
 Dept of Management and Information System, Hyupsung University

요 약

프로그램 개발 단계에서 프로그램을 수정할 경우, 전체 프로그램을 다시 평가하는 대신 수정한 부분과 그 부분에 영향 받는 부분만을 다시 평가하는 것이 점진 평가 방법이다. 본 논문에서는 의미 구조에 직접적으로 영향을 주는 변수의 값을 나타내는 속성을 중심으로 종속성을 표시하고, 여러 프로세서에서 병렬로 처리하는 대신 멀티쓰레딩 기능을 활용하여 객체 지향언어인 자바 언어에서 점진 평가를 동시에 효율적으로 수행할 수 있는 병행 점진 평가 방법을 제시하고 모의실험을 통해 병행 점진 평가 방법의 효율성을 분석한다.

1. 서론

하드웨어의 성능이 향상됨에 따라 멀티쓰레딩을 활용하여 병행 처리하는 연구가 다양하게 진행되고 있다. 프로그램 개발 단계에서 프로그램을 수정할 경우, 점진 평가는 전체 프로그램을 다시 평가하는 대신 수정한 부분과 그 부분에 영향 받는 부분만을 다시 평가하는 방법이다. 점진 평가를 동시에(concurrent) 수행하기 위해 여러 프로세서에서 병렬로 처리하는 대신 멀티쓰레딩 기능을 활용하여 점진평가의 효율성을 증대시키는 연구가 필요하다.

본 논문에서는 종속 차트(dependency chart)[2]를 사용하여 객체 지향언어인 자바 언어에서 병행 점진 평가를 동시에 수행할 수 있는 방법을 제시한다. 의미 구조에 직접적으로 영향을 주는 변수의 값을 나타내는 속성을 중심으로 종속성을 표시하여 병행 점진 평가를 수행한다. 특히, 자바의 멀티쓰레딩 기능을 활용하여 병행 점진 평가를 수행하고 실험을 통해 제시된 병행 점진 평가의 효율성을 분석한다.

2. 멀티쓰레딩 분석

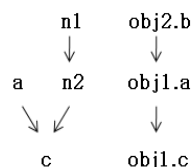
하드웨어 성능의 향상으로 멀티쓰레딩 기능을 활용하면 점진 평가를 좀 더 효율적으로 수행할 수 있다. 효율적인 병행 점진 평가를 수행하기 위해, 점진 평가를 수행하는 과정 중 멀티쓰레딩이 가능한 부분을 찾아 분석하는 과정이 필요하다. 그림 1의 자바 프로그램을 통해 멀티쓰레딩이 가능한 경우를 분석하면 멀티태스킹방식으로 병행 점진평가를 수행할 수 있다.

```
public class Multitask {
  int a = 10, b = 15, c;    --- ①
  ...
  void methodA() {
    int n1 = 1, n2;        --- ②
    n2 = n1 + 10;          --- ③
    c = n2 * a;            --- ④
```

```
...
}
...
public static void main(String args[]) {
  Multitask obj1, obj2;
  obj1 = new Multitask ();
  obj2 = new Multitask();
  ...
  obj1.methodA();         --- ⑤
  obj1.a = obj2.b * 2;    --- ⑥
  obj1.c = obj1.a;        --- ⑦
  ... }
}
```

(그림 1) 자바 프로그램

Multitask 객체 obj1과 obj2 는 객체 속성 변수로 a, b, c를 갖고 지역변수로 n1, n2를 갖고 있다. 각 변수의 종속성을 표현한 종속 링크만을 갖는 종속차트는 그림 2와 같다.



(그림 2) 종속차트의 종속링크

그림 1의 ⑤번 명령문이 먼저 실행된 후 ⑥번과 ⑦번 명령문이 실행되므로 그림 2와 같은 종속 링크가 만들어지게 된다. 그림 2에서 볼 수 있듯이 두 개의 링크 그래프로 나뉘는데 각 링크 그래프에 소속된 변수 값이 변경될 때 각각을 평가하는 과정은 멀티태스킹이 가능하다. 두 링크 그래프 안에 소속된 변수들의 값의 연관성이 없어 동시에 작업해도 오류 없이 정확한 값을 갖게 된다. 왼쪽 그래프의 경

우에도 c의 상위 정점(vertex)에 있는 두 부분그래프에 소속된 변수들의 값의 변화에 대해서도 멀티태스킹이 가능함을 알 수 있다. a와 n1의 경우 두 변수에 대해 동시에 병행 점진 평가를 수행해도 서로의 값에 영향을 주지 않기 때문이다. a와 n2의 경우도 병행 점진 평가가 가능한 변수들이다. a와 n1이나 a와 n2처럼, 병행 점진 평가가 가능한 변수들을 독립적인 변수라 하고 병행 점진 평가가 불가능한 n1이나 n2는 종속적인 변수이다.

3. 병행 점진 평가 방법

점진 평가를 수행하려면, 변수 값을 수정하는 명령문이 나올 경우 종속차트의 종속링크를 통해 그 변수 값에 영향 받아 변하게 될 변수들을 깊이우선탐색(DFS)으로 추적하여 영향 받는 명령문들을 다시 평가한다. 변수 값이 변할 때 점진 평가를 수행하기 위해 종속링크를 찾아 가는 것이 중요하다. 만일 여러 변수들에 변화가 생겼을 경우 각 변수들에 대해 동시에 종속 링크를 추적하여 영향받는 변수들을 찾아 change-set[2]를 구성하고 change-set에 속한 변수들 중에서 프로그램이 실제로 실행될 때 반드시 평가되는 변수들인 필수 변수들로 evaluated-set[2]를 구성한다. evaluated-set에 속한 변수들을 다시 평가하게 되는데 독립적인 변수의 경우 멀티태스킹이 가능하므로 동시에 병행 평가하고 종속적인 특성을 갖는 변수들은 순차적으로 평가하게 된다. 평가하는 방법은 종속차트에 있는 정보를 검색하여 그 변수들이 속해있는 명령문들을 평가하게 된다. 따라서, 모든 변수들을 순차적으로 평가하는 것보다 변수의 특성에 따라 동시에 병행 점진 평가를 수행할 수 있으므로 효율적인 병행 점진 평가를 수행할 수 있다.

자바에서 점진 평가로 고려할 수 있는 경우는 지역 변수의 변화, 객체 속성 변수의 변화, 객체 이름의 변경, 클래스 이름이나 메서드 이름의 변경 등이다[2]. 멀티태스킹의 효율성을 높이기 위해 지역변수와 객체속성 변수의 변화를 제외한 나머지 경우의 변화에 대해서는 미리 변경될 사항을 종속차트에 조정하는 것이 더 바람직하다. 그림 1의 ⑥번 명령문 `obj1.a = obj2.b * 2;` 에서 `obj1.a` 대신 `obj2.a`로 객체 이름을 변경하면 해당 객체의 속성 변수에 변화가 생기게 되고 종속 링크도 변경해서 객체 `obj2`의 `a`값이 바뀌도록 종속차트를 조정한다. 클래스이름이나 메서드이름이 변경될 경우 해당 클래스나 메서드를 종속차트에서 찾아 소속 변수들의 클래스이름이나 메서드이름을 변경한다.

4. 병행 점진 평가의 효율성 분석

병행 점진 평가의 효율성을 분석하기 위해 자바로 작성된 네 가지 유형의 프로그램에 대해 모의실험을 통해 병행 점진 평가의 효율성을 분석한다. 합계와 평균을 계산하는 프로그램(P1), 최대값과 최소값을 찾는 프로그램(P2), 급료를 계산하는 프로그램(P3), 배열로 선형 리스트를 구현하는 프로그램(P4)에 대해 실험하고 네 가지 명령문인 배경문, 조건문, 반복문, 메서드 호출문에 대해 변수 값을 수정하였고 그 수정으로 점진 평가하는 명령문의 비율과 병행 점진 평가를 수행하는 명령문의 비율을 비교하여 수정하는 명령문 유형에 따라 병행 점진 평가의 효율성을 분석한다.

프로그램 유형별로 비교했을 때 P4 프로그램의 경우 모든 수정 유형을 평균적으로 분석해 볼 때 점진 평가보다 1.7배 빠르게 실행되어 다른 프로그램 유형보다 더 효율적이다. P4의 경우 여러 수정 유형에서 다른 프로그램들보다

독립적인 변수들이 많고 그 변수들에 대해 수정될 경우 병행 점진 평가가 가능하므로 다른 프로그램 유형보다 효율적이라는 사실을 알 수 있다. 네 가지 명령문 유형의 병행 점진 평가의 수정 중 호출문의 경우 멀티태스킹의 가능성이 높은 독립적인 변수들의 수정 가능성이 높아서 병행 점진 평가가 점진 평가보다 1.6 배 빠르게 수행되고 다른 수정 유형보다 효율성이 높음을 알 수 있다.

평균적으로 고려해볼 때 독립적인 변수들의 수정 가능성이 높을 경우 점진 평가보다 병행 점진 평가를 수행했을 때 평가 효율성이 높음을 알 수 있고 점진 평가하는 것을 100%로 놓았을 때 병행 점진 평가를 수행했을 때 점진 평가의 60-70% 만 실행하여 1.45배 빠르게 실행되어 점진 평가보다 효율적인 병행 점진 평가를 수행하게 된다.

<표 1> 프로그램 점진 평가 결과 (단위:%)

프로그램	배경문		조건문		반복문		호출문		평균	
	점진	병행	점진	병행	점진	병행	점진	병행	점진	병행
P1	30	21	12	10	31	24	35	23	27	19
P1	89	62	3	2	90	68	91	59	68	48
P2	33	24	32	26	38	29	41	27	36	27
P2	93	66	94	75	95	72	96	62	95	69
P3	6	4	7	4	76	58	71	40	40	27
P3	8	6	13	19	88	67	82	47	48	32
P4	9	5	17	9	88	58	81	46	49	30
P4	5	3	9	5	89	59	83	47	47	28
평균	34	24	23	18	74	54	73	44	51	35

5. 결론

본 논문에서는 병행 점진 평가를 수행하기 위해 종속 차트[2]를 사용하여 객체지향언어인 자바에서 멀티쓰레딩을 활용하여 병행점진평가를 수행하는 방법을 제시하였다.

네 가지 프로그램 유형에 대해서 실험하였고 네 가지 명령문인 배경문, 조건문, 반복문, 메서드 호출문에 대해 수정한 후 다시 평가될 명령문 수를 중심으로 병행 점진 평가의 효율성을 검토해 보았다. 실험 결과 멀티태스킹의 가능성이 높은 독립적인 변수들의 수정 가능성이 높은 경우 병행 점진 평가를 수행하는 것이 점진 평가를 수행하는 것보다 더 효율적이었다.

평균적으로 병행 점진 평가를 수행했을 때 점진 평가의 60-70%만 실행하여 병행 점진 평가가 1.45배 빠르게 수행되어 병행 점진 평가가 점진 평가 보다 더 효율적으로 수행된다는 사실을 확인할 수 있다.

참고문헌

- [1] 한정란 "작용 식 기반 점진 해석" Ph. D Thesis 이화여대 1999.
- [2] 한정란 "확장된 종속차트를 사용한 효율적인 점진 평가 방법" 인터넷정보학회 논문지, 제10권 2호, pp.75~84, 2009.
- [3] 한정란 "객체 지향 언어를 위한 의미 명세" 인터넷정보학회 논문지, 제8권 5호, pp.35~43, 2007.
- [4] Han junglan "Action Equations for Object Oriented Programming Language" In Proceeding of ICUT pp.101~108, 2007.